105074
#1

Experimental and Computational Studies of Rotors in Forward Flight and Descent

Final report

Submitted to

Dr. Yung Yu
yhoon@konkuk.ac.kr
Professor, Aerospace Engineering
Konkuk University
Seoul 143-701, Republic of Korea

Prepared by
Dr. Daniel P. Schrage
daniel.schrage@ae.gatech.edu

Professor and Director
Center of Excellence in Rotorcraft Technology
Center for Aerospace Systems Engineering

J. V. R. Prasad
Jvr.prasad@ae.gatech.edu
Professor

Lakshmi N. Sankar
lsankar@ae.gatech.edu
Regents Professor and Associate Chair
School of Aerospace Engineering
Atlanta, GA 30332-0150

February 23, 2007

INTRODUCTION

The researchers at the School of Aerospace Engineering (Prof. Daniel P. Schrage, Prof. Lakshmi N. Sankar, and Prof. J. V. R. Prasad) are pleased submit collaborative research activities in the area of rotorcraft technology. Work was done in the following three areas.

- Experimental and Computational Studies of Rotors in Forward Flight and Descent
- Integrated Autonomous Behaviors of UAVs with Envelope Protection
- An Integrated Product and Process Development (IPPD) Approach for Rotorcraft Preliminary Design Optimization

This report gives the progress made in these three areas. To improve readability, the progress report is given in three separate sections.

Appendix I

I. Experimental and Computational Studies of Rotors in Forward Flight and Descent

Research Objectives

The objective of this research is to jointly develop first principles based analyses for modeling rotors in hover, forward flight, and descent. It is anticipated that these jointly developed tools will be used to design next generation rotor blade sections as well as rotor blades themselves, and will be useful in joint studies of the HART-III workshop test cases.

Progress report for the First Year

A CFD code named GENCAS(Generic Numerical Compressible Airflow Solver) has been developed by a graduate researcher (Byung Young Min) and used as a test bed for the easy assessment of baseline algorithms and advanced turbulence models. GENCAS is a 2D/3D, multi-block, compressible and Euler/N-S CFD code. It has 4^{th} order Runge-Kutta explicit and LUSGS implicit time marching schemes. Roe's FDS and AUSMPW+ schemes are implemented for spatial discretization. MUSCL reconstruction scheme is used up to 3^{rd} order. 5^{th} order and 7^{th} order WENO scheme is currently available and being tested. For turbulence models, S-A (Spalart-Allmaras), standard k-ε, Wilcox's k-ω, Menter's k-ω SST, and DES model for each are currently available. KES, one of the most advanced turbulence model, will be added in the future.

Validation studies have been done for NACA0012 airfoil, RAE2822 airfoil and Goldman turbine vane cases.

Validation 1: NACA0012

Inviscid calculations of NACA0012 airfoil at two different angles of attack have been done and compared with AGARD standard airfoil database to validate baseline spatial discretization and time marching schemes. Figure 1 shows the grid used for this study.

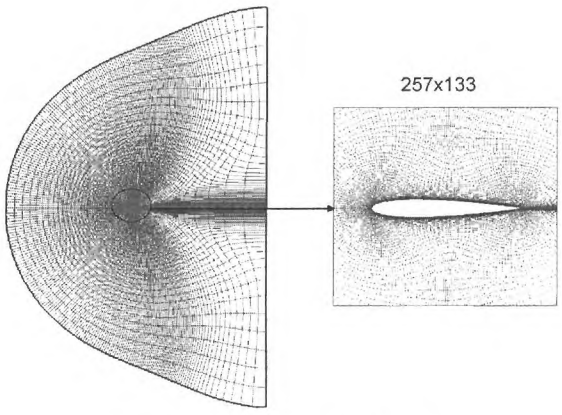


Figure 1. NACA 0012 grid

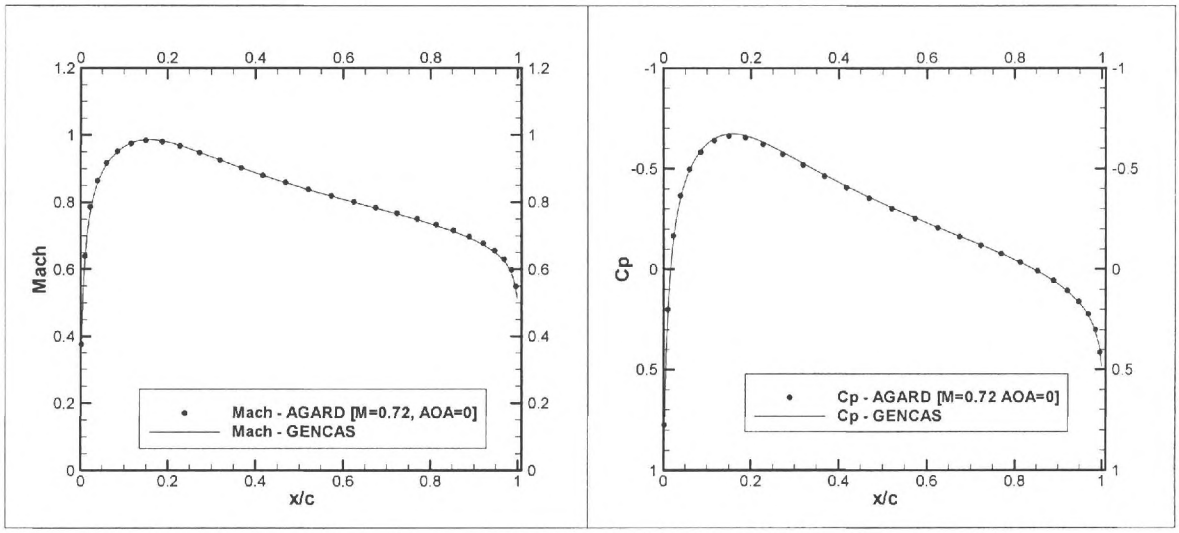


Figure 2. Surface Mach number and pressure coefficient comparison (M=0.72, AOA=0)

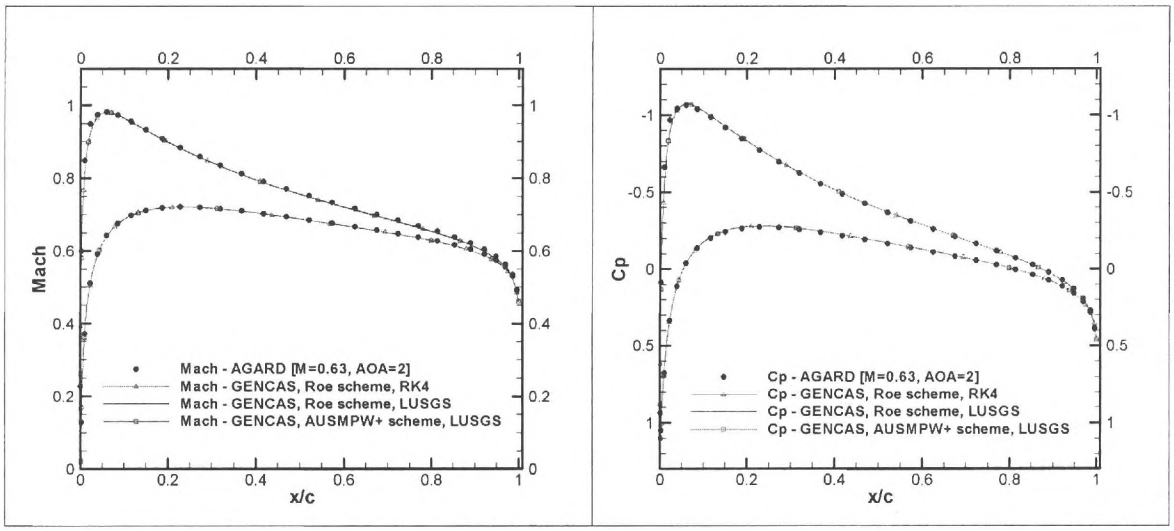


Figure 3. Surface Mach number and pressure coefficient comparison (M=0.63, AOA=2 deg.)

Figure 2 and 3 show comparison of surface Mach number and pressure coefficients. It is seen that the results from GENCAS compare well with the AGARD standard test base data.

Validation 2: RAE2822

To check the validity of turbulence models in GENCAS, a well known RAE2822 transonic airfoil case was computed and compared with experimental result. Freestream Mach number is 0.75, angle of attack is 2.72 degree, and Reynolds number is 6.2×10^6.

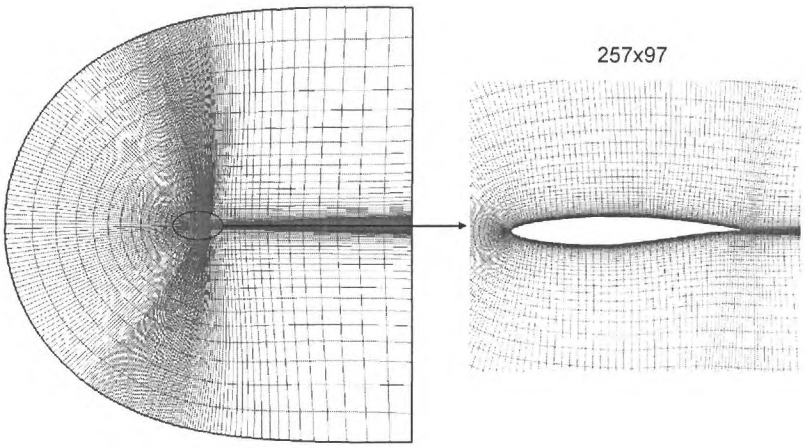
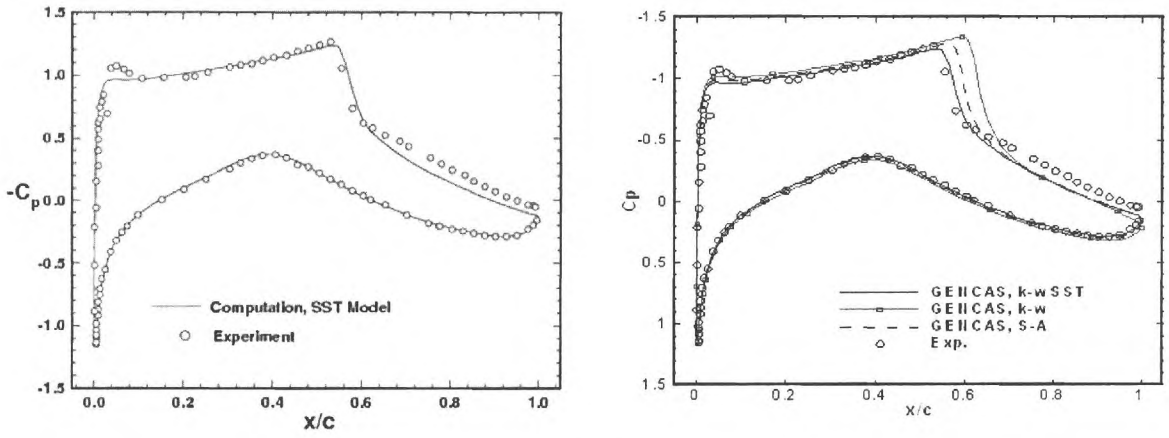


Figure 4. RAE2822 airfoil grid

Figure 4 shows grid used for this study, and figure 5 shows comparison of computed result from CFL3D and GENCAS with experimental data. CFL3D computational result and experimental data were taken from CFL3D Manual (Version 5.0). In this case, the SST model from both CFL3D and GENCAS predicts the shock position well. It was found that the Spalart-Allmaras model predicted slight off to downstream and that the k-ω model was significantly off compared to experimental data and he other models. For this reason, it is recommended that the S_A model be sued for preliminary engineering studies, and that the more costly but accurate k-W-SST model be used for the final stages of rotor analyses and design studies.



(a) CFL3D (b) GENCAS

Figure 5. Surface pressure coefficient comparison

Validation 3: Goldman Turbine Vane

Various turbulence models were tested and compared with experimental data of stationary Goldman turbine vane. Figure 6 shows grid of one turbine vane. C-H topology was used. Periodic boundary condition was applied to upper and lower outer boundaries. Design pressure ratio is 0.6705, inlet Mach number is 0.211, and Reynolds number is 1.73×10^5.

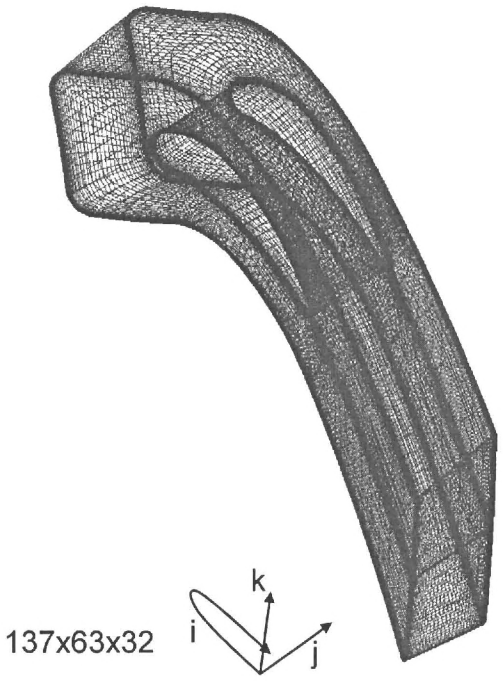


Figure 6. Goldman turbine grid

Figure 7 shows pressure ratio comparison from different turbulence models with experimental data at midspan surface on the vane. S-A model was worst in predicting trailing edge separation, but S-A DES, k-ω and k-ω SST are all in relatively good agreement with experiment.

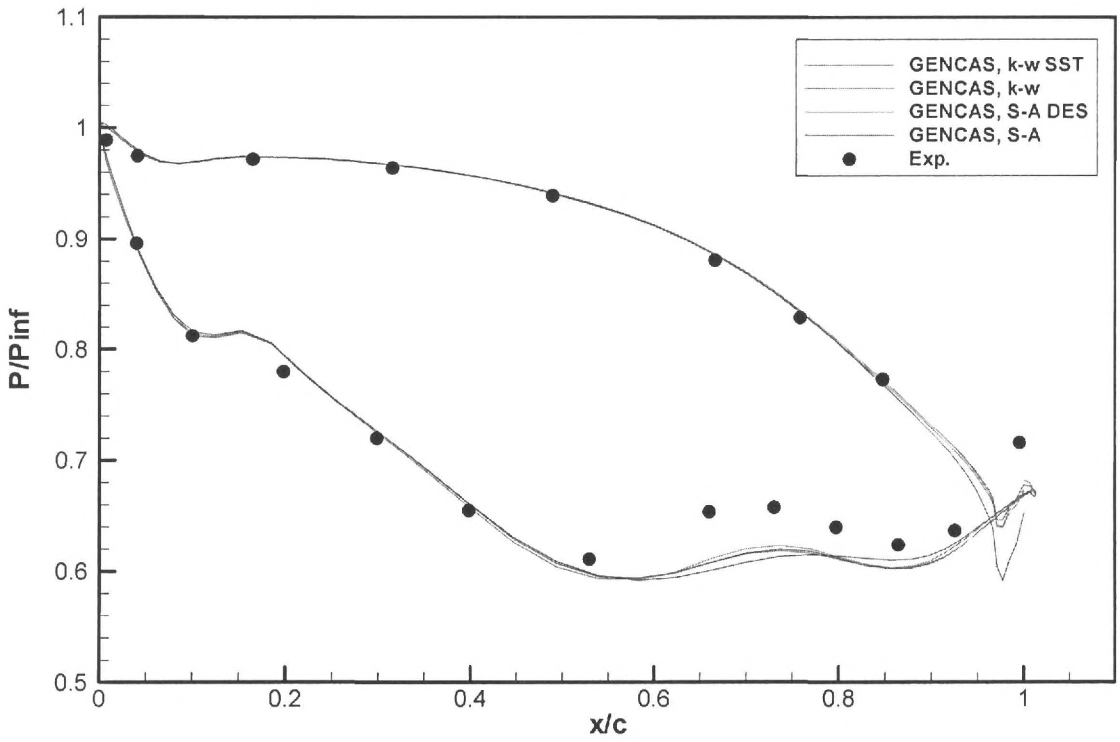


Figure 7. Pressure ratio comparison

HART-I Test Data: Calculations are being done using GENCAS for one of the three HART-I test cases (baseline). At this writing, GENCAS has not been coupled with a CSD model. It is recommended that GENCAS simulations be coupled with a CAMRAD-II based model of the HART rotor to include the blade elastic deformations and trim information. When this is completed, it is anticipated that results similar to figures 8 and 9 shown below (done using Ga Tech GT-Hybrid code coupled to RCAS) will be obtained. The present researchers plan on working with the researchers at Konkuk University in coupling GENCAS to CAMRAD-II, and applying the coupled solver to the HART-I, HART-II, and HART-III cases.

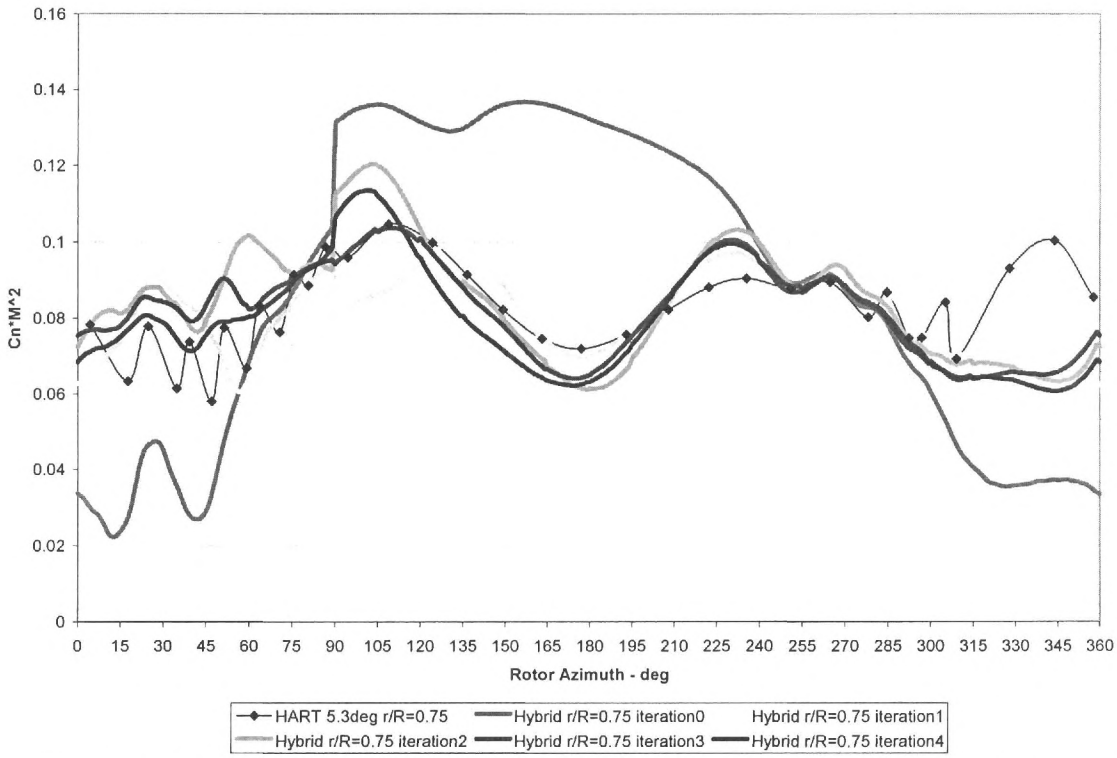


Figure 8. Convergence of Coupled CFD/CSD Loads at 75%R and Comparisons with Experiments (Ga Tech Hybrid Code, CFD+CSD)

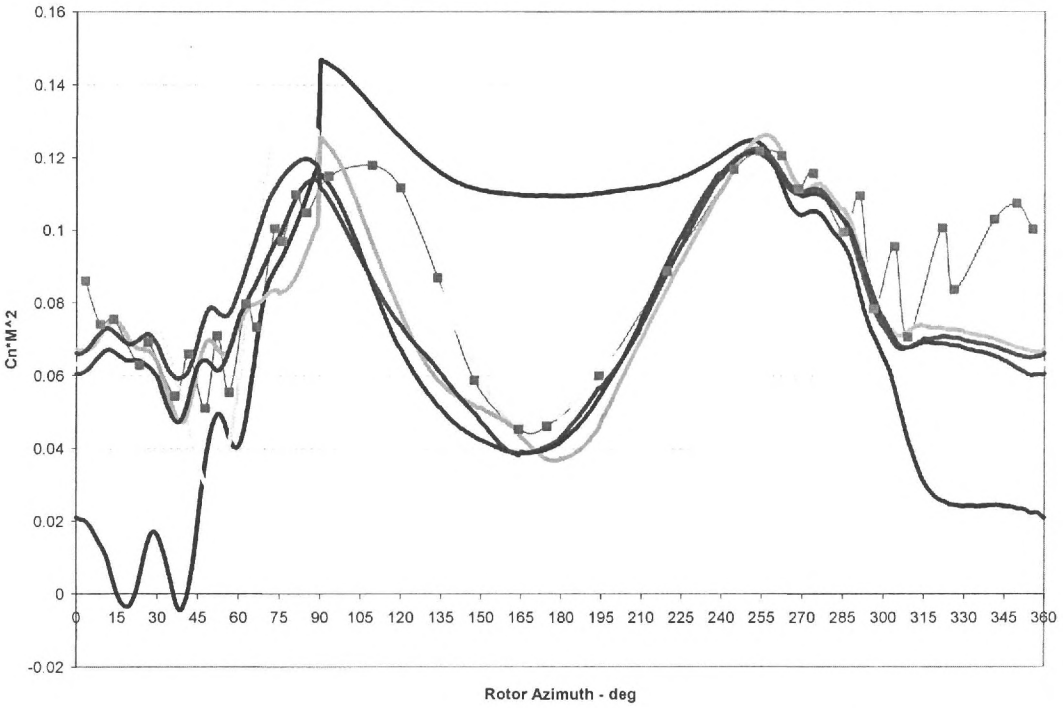


Figure 9. Convergence of Coupled CFD/CSD Loads at 87%R and Comparisons with Experiments (Ga Tech Hybrid Code, CFD + CSD)

APPENDIX II

INTEGRATED AUTONOMOUS BEHAVIORS OF UAVs WITH ENVELOP PROTECTION

Year 1 Project Progress Report

J.V.R. Prasad, Professor
Jongki Moon, Ph.D. Student
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

November 15, 2007

SUMMARY

The main focus of the year 1 study is to combine path planning for obstacle avoidance with envelop protection, to ensure that UAV is operated within its safe operational limits. In this integrated framework, distance to obstacles along the LOS is treated as an external limit parameter, while the operational envelope is considered as an internal limit parameter. Information on obstacles, such as location and size, is unknown until detected. An adaptive neural net based estimation is used for limit parameter dynamics. Since this study focuses on a time optimal solution in this integrated approach, the proposed method calculates and applies maximum available control command so that it can avoid obstacles. Numerical examples for load factor protection using GTMax shows that using the proposed approach, it is possible to determine collision free trajectories without violating a load factor limit boundary.

INTRODUCTION

The advantages of unmanned aerial vehicles (UAVs) in the aviation arena have led to extensive research activities on autonomous control of UAVs to achieve specific mission objectives. The desired objective of the UAV could be commanded by the operator or pre-programmed without human intervention. This objective may be to go to a single point, pass through several way-points, avoid obstacles, or track a target, etc. [1]. An essential part of vehicle autonomy consists of a trajectory planning and guidance system that enables it to safely maneuver through a particular environment. This environment may contain obstacles and zones that the vehicle is not allowed to enter and may not be fully characterized at the start of a mission. Obstacles may be detected as the vehicle moves through the environment or their location may change over time. During an obstacle-avoidance maneuver that may require aggressive maneuvering, it is imperative that the flight envelope of a UAV is protected to ensure safety and structural integrity of the vehicle. Previous work has considered these two issues, i.e., obstacle avoidance and envelope protection, separately without a consideration of the tight coupling that exists between these two. For example, if aggressive maneuvering is needed during the course of obstacle avoidance, it may result in excessive commands to the UAV's flight control system so as to result in exceeding its safe flight envelope. Thus, there is a need to develop methods that combine obstacle avoidance and envelope protection into a single framework.

Envelope protection is the task of monitoring and maintaining vehicle operation within an operational envelope. The operational envelope is determined by system variables, called as limit parameters. Ensuring that vehicle operation remains within the operational flight envelope translates into maintaining the limit parameter response within the limit boundaries. Several methods for envelope protection in rotary-wing UAVs have been proposed. Since maneuverability constraints for the comfort of a human pilot are no longer applicable to UAVs, controller design can be focused solely on meeting mission needs. In addition, UAVs are expected to be operated more aggressively than their manned counterparts, closer to its limit boundaries. However, the task of envelope protection must be done automatically due to the absence of a pilot. Some of the past studies have made use of artificial neural networks [6, 7] and dynamic trim method [8] in the prediction of vehicle operations close to its limit boundaries.

Recent studies [9, 10] show interesting results of reactive obstacle avoidance for autonomous helicopters. However, limits on the vehicle operational envelope have not been considered in these studies. On the other hand, reactionary envelop protection method has been proposed and successfully evaluated in flight [11, 12]. The reactive envelope protection method makes use of a procedure which is somewhat similar to that of obstacle avoidance. In the case of envelope protection, the limit boundaries are treated as internal obstacles that the vehicle must avoid, similar to an obstacle may be treated as an external limit boundary the vehicle must avoid.

This report is arranged as follows: Section 2 explains obstacle avoidance technique and envelope protection. An integrated framework for simultaneous obstacle avoidance and envelope protection is discussed in section 3. Numerical simulation results are presented in section 4, and section 5 provides conclusions followed by recommendations for further work.

Obstacle avoidance

In general, guidance problem can be decomposed into three levels of functions: far field, mid field, and near field. These three functions can be realized naturally in terms of the three feedback loops shown in Fig. 1.

A classical approach to solving the obstacle-avoidance problem would be to solve an optimal control problem. For example, the standard indirect, direct, and homotopy methods for the numerical solution of optimal control problems may be used as shown in references [2, 3]. These methods are generally too slow to be used in real time. One of the modern approaches is to use differential geometric techniques [4, 5]. These methods depend on finding special outputs which are called flat outputs. Using these outputs, the complete differential behavior of a system can be found in terms of outputs and their derivatives. However, most of these techniques do not take into account actuation and state constraints.

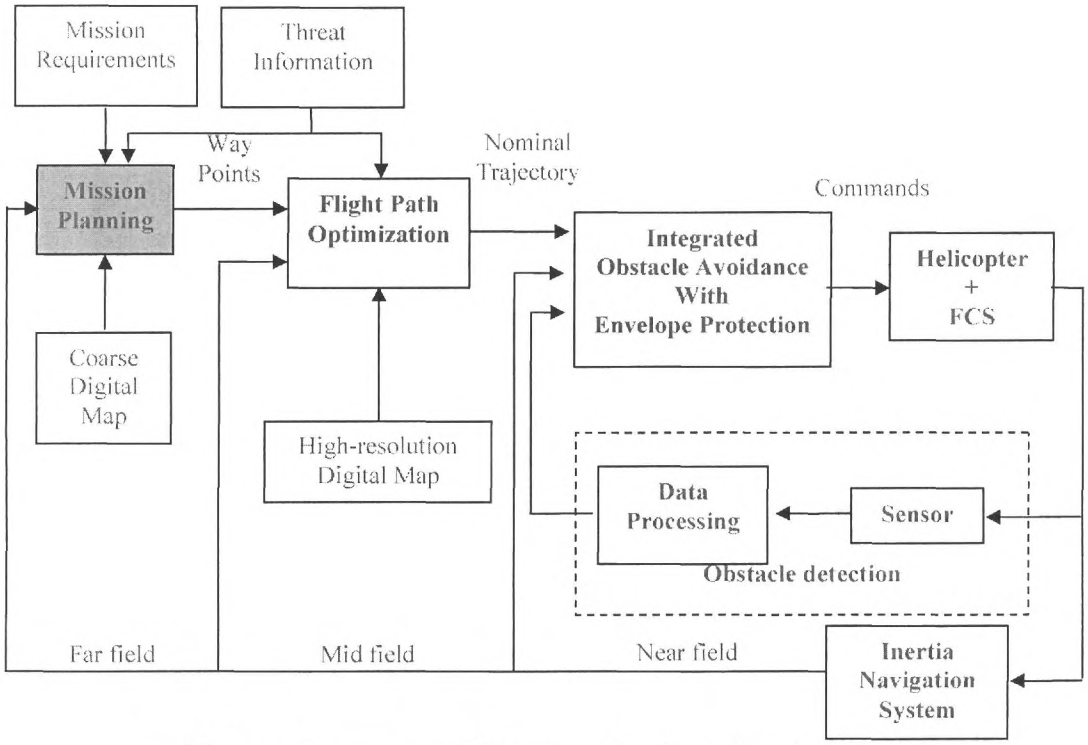


Fig. 1 Autonomous guidance structure for helicopter UAVs

Envelope protection

Recent work on envelop protection proposed two new approaches, viz., a reactionary envelope protection approach [11, 12] and a nonlinear trajectory generation (NTG) based envelop protection approach [13]. In the NTG approach, an optimal formulation combined with differential geometric methods is used to arrive at control/command margins available that, if applied, will take the vehicle response to its limit boundaries. Actual control/command inputs are limited by the control/command margins in order to protect the vehicle from exceeding its limit boundaries. In the reactionary approach, a safe limit parameter response is arrived at using a geometric approach similar to obstacle avoidance and the nominal control/command inputs are modified to track a safe limit parameter response profile.

Either of the above approaches requires a model of limit parameter dynamics. Reference 7 shows that an adaptive architecture (see Fig. 3) can be used to obtain an estimate of limit parameter dynamics:

$$\hat{y}_p^{(r)} = \hat{h}\left(\hat{y}_p, \hat{y}_p^{(1)}, \cdots, \hat{y}_p^{(r-1)}, u_p\right) + \nu_{ad}(\mu) - \nu_{dc} \qquad (1)$$

where $\hat{y}_p$ is an estimate of limit parameter, $\hat{h}(\cdot)$ is an approximation to the unknown limit parameter dynamics, r is the relative degree of the limit parameter dynamics, ν_{ad} is the output of neural networks used here to compensate for the error in the approximation to the limit parameter dynamics, and ν_{dc} is the output of a dynamic compensator. Since we can assume a linear approximation to $\hat{h}(\cdot)$, Eq. (1) can be rewritten as follows:

$$\hat{y}_p^{(r)} = \sum_{i=0}^{r-1} a_i \hat{y}_p^{(i)} + b u_p + \nu_{ad}(\mu) - \nu_{dc} \cdot \qquad (2)$$

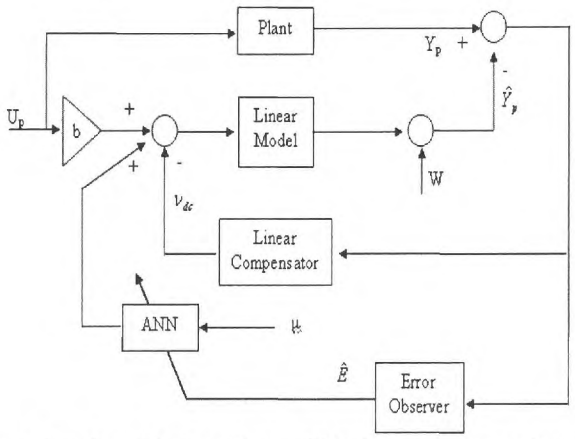


Fig. 3 Adaptive Estimation of limit parameter dynamics

A single hidden layer neural network is often used for the adaptive element ν_{ad}. The weights of NN are updated on-line according to the weight update law, which is given by Eq. (2) to capture the modeling uncertainty resulting from the linear approximation

$$\dot{\hat{V}} = -\Gamma_N \left[\mu \hat{E}^T PB \hat{W}^T \sigma' + \kappa \left\| \hat{E}^T PB \right\| \hat{V}\right], \qquad \dot{\hat{W}} = -\Gamma_N \left[\hat{\sigma} \hat{E}^T PB + \kappa \left\| \hat{E}^T PB \right\| \hat{W}\right] \qquad (3)$$

In Eq. (3), $\hat{M}$ and $\hat{N}$ are referred to as the output and hidden layer weight matrices, respectively. In addition, Γ_M and Γ_N are the NN learning rates of the output and hidden layers, respectively, and κ is known as the e-modification gain.

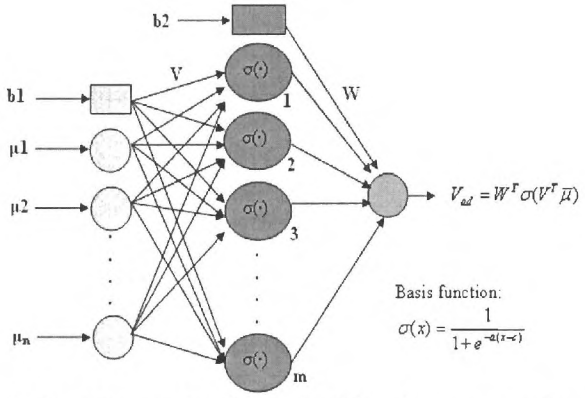


Figure 4. Structure of a single hidden layer neural network

Also, the matrix P is the solution to the following Lyapunov equation:

$$A^T P + PA = -Q, \tag{4}$$

where Q is a positive definite matrix, and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 & a_1 & \cdots & \cdots & \cdots & a_{r-1} \end{bmatrix}. \tag{5}$$

The approximate linear model in Eq. (2) can be selected to be stable. Thus, the parameters in Eq. (5) are chosen so that matrix A is Hurwitz and $B \in R^r$ is a r-dimensional unit vector with its last element equal to 1.

In the reactionary envelop protection approach, the nominal control input is modified so that the limit parameter response is able to follow a safe-response profile (y_s) which is generated using a geometric technique [11, 14]. The nominal control/command input is modified to track the safe response profile arrived at using the geometric technique. The control/command correction needed is [11]

$$u_{corr} = \left[y_s^{(r)} - \left(\sum_{i=0}^{r-1} a_i y_s^{(i)} + \frac{\partial \hat{h}_r}{\partial u} u_p + \nu_{ad} - \nu_{dc} \right) \right] \left(\frac{\partial \hat{h}_r}{\partial u} \right)^{-1} \tag{6}$$

The corrective input given by Eq. (6) is based on the adaptive estimation of the limit parameter dynamics and guarantees tracking of safe-response profile. How well the adaptive NN can approximate the modeling uncertainty will determine the overall tracking performance.

INTEGRATED LIMIT AVOIDANCE

Integrated Approach

An integrated approach to obstacle avoidance and envelope protection may be arrived at by treating obstacles as external limit boundaries. In this context, one can use the NTG approach wherein the external limit boundaries (obstacles) as well as the internal limit boundaries are treated simultaneously as constraints. Similarly, in the reactionary approach, safe response profiles associated with avoidance of external limit boundaries (obstacles) as well as avoidance of internal limit boundaries can be arrived at. However, modifications to the nominal control/command in order to track a safe response profile is not so obvious here because the safe response profile associated with obstacle avoidance may conflict with the safe response profile associated with limit avoidance. In this study, an initial attempt is made at addressing this issue by considering a simpler 2-D problem of simultaneous obstacle avoidance and limit avoidance by a UAV flying at constant speed.

A helicopter UAV is equipped with radar with a maximum range r_{max} and a field of view (FOV) 2ϕ. The radar cone is assumed to point along the direction of the vehicle velocity. An obstacle detected with such a radar at some position **r** relative to the vehicle is detected as being along the velocity vector **V**, i.e., along $\mathbf{r}_0$ in Fig. 5.

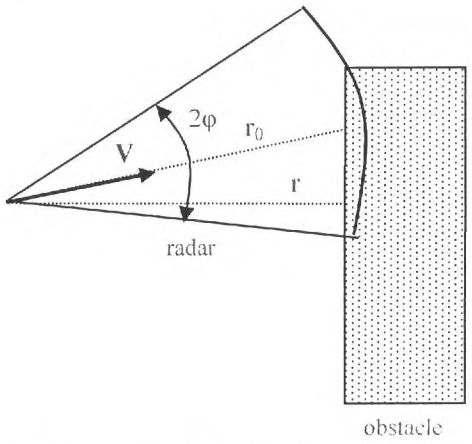


Fig. 5 Schematic model of UAV with radar

The following section describes a discrete time simulation that was performed to gain further insights into the performance of the nonlinear guidance law for obstacle avoidance. Consider Fig. 6 showing the evolution of the guidance logic in one time step increment.

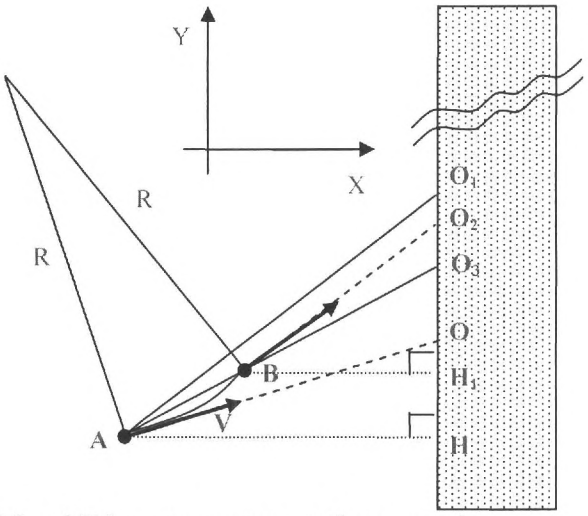


Fig. 6 Discrete representation: one time step

Property 1: Using one time step simulation, we can obtain a relationship between pitch rate and the time rate of relative distance along the LOS during a constant speed avoidance maneuver as follows:

$$\dot{r}_0 = -V + r_0 \dot{\theta} \tan\theta . \tag{7}$$

A proof of this property is given in the Appendix.

Remark) If a pitch rate is commanded as 0, then a closing rate is equal to the UAV speed. This is illustrated in Fig. 7

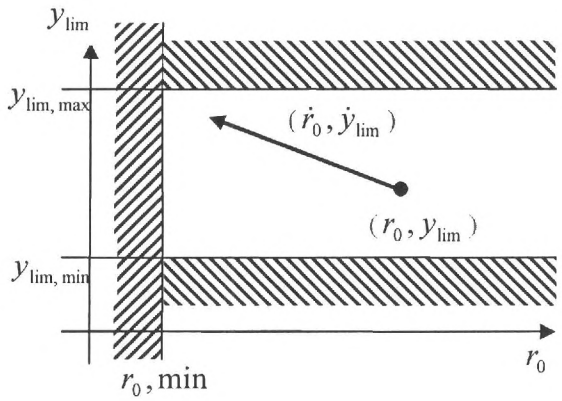


Fig. 7. Geometric illustration of Eq. (7).

3.2 Minimum Principle

We first define the performance index J. For a minimum time problem:

$$J = \int_0^T L\, dt = \int_0^T 1\, dt = T \tag{8}$$

The kinematics and the control/command (pitch rate here) are adjoined to the performance index with multipliers λ_1, λ_2 and λ_3, to form the Hamiltonian H:

$$H = L + \lambda^T f = 1 + \lambda_1 V \cos\theta + \lambda_2 V \sin\theta + \lambda_3 u \tag{9}$$

where the dynamics of the multipliers satisfy

$$\dot{\lambda} = -\frac{\partial H}{\partial X} = -\begin{bmatrix} 0 \\ 0 \\ -V\sin\theta + V\cos\theta \end{bmatrix} \tag{10}$$

By defining the Hamiltonian and the multiplier dynamics in this way, the minimum principle requires that on the optimal trajectory (X^*, λ^*), the optimal pitch rate u^* must minimize H over the set of all possible inputs u_a at each instance in time:

$$H(X^*, u^*, \lambda^*, t) \le H(X^*, u, \lambda^*, t) \qquad \forall u \in u_a,\ t \in [0, T] \tag{11}$$

Because λ_1 and λ_2 are constant, satisfying the condition in Eq. (11) is equivalent to satisfying

$$\lambda_3 u^* \le \lambda_3 u \quad \forall u \in u_a \tag{12}$$

The necessary condition of Eq. (12) results in the following properties of the optimal path:

Property 2: The optimal path consists of maximum pitch rate available and straight lines.
Proof: Because λ_1 and λ_2 are constant, $\dot{\lambda}_3$ is equal to zero only for a specific pitch angle θ_c, which satisfies $\lambda_1 \sin\theta_c = \lambda_2 \cos\theta_c$. Thus during any nonzero time interval, λ_3 is constant if and only if the UAV travels in a straight line parallel to the characteristic direction, θ_c. The optimality condition in Eq. (12) allows only three input values: on any segment where $\lambda_3 > 0$, $u^* = q_{\min}$; on any segment where $\lambda_3 < 0$, $u^* = q_{\max}$; on any segment where $\lambda_3 = 0$, UAV must travel in a straight line, implying $u^* = 0$.

Property 3: All straight line segments and changes in pitch rate on the optimal path must occur on a single line.
Proof: the equation for $\dot{\lambda}_3$ becomes $\dot{\lambda}_3 = \dot{y}\lambda_1 - \dot{x}\lambda_2$, which integrates to
$\lambda_3 + x\lambda_2 - y\lambda_1 = c$

where c is a constant. Because c, λ_1, and λ_2 are constant, each value of λ_3 defines a line parallel to the characteristic line. The line defined by $\lambda_3 = 0$ is the line on which all switching and straight-line travel must occur.

Determination of Control Command

Since we also consider limit parameter protection, the available control input is not a constant value, but restricted by the limit parameter dynamics. Therefore, we need to calculate how much pitch rate command is available at each instance. In order to construct an estimate of available control/command input for a limit boundary, the following equation can be used

$$u_{avail} = \left[y_d^{(r)} - \left(\sum_{i=0}^{r-1} a_i y_d^{(i)} + \frac{\partial \hat{h}_r}{\partial u} u_p + \nu_{ad}(\bar{\mu}) - \nu_{da} \right) - k(\hat{y}_p - y_d) \right] \left(\frac{\partial \hat{h}_r}{\partial u} \right)^{-1} \quad (13)$$

by choosing appropriate values for the desired limit parameter and its derivatives $\left(y_d, y_d^{(1)}, \cdots, y_d^{(r)}\right)$. These variables are set to pre-determined values based on the acceptable limit parameter response aggressiveness, which means that the available control/command input is based on the notion that the limit parameter response should slow down not to violate its limit boundary. Thus, in case of relative degree one (r=1) limit parameter response, for example, the safe profile can be set as

$$\left[y_d, y_d^{(1)} \right] = \begin{cases} \left[\bar{y}_{\lim}, \dfrac{\bar{y}_{\lim} - \hat{y}_p}{\tau} \right] & \text{if } \left| \hat{y}_p - \bar{y}_{\lim} \right| > \varepsilon \\ \left[\bar{y}_{\lim}, 0 \right] & \text{if } \left| \hat{y}_p - \bar{y}_{\lim} \right| \le \varepsilon \end{cases} \quad (14)$$

where $\bar{y}_{\lim}$ and ε are upper bound of the limit parameter and tolerance of limit boundary, respectively. The particular choice for y_d and $y_d^{(1)}$ in Eq. (14) corresponds to the case wherein one desires to use the full flight envelope available without violating the limit parameter boundary. After defining the estimate of tracking error, $\hat{e}_d = \hat{y}_p - y_d$, then the available control/command can be rewritten as

$$u_{avail} = \left(-\hat{e}_d^{(r)} + \sum_{i=0}^{r-1} a_i \hat{e}_d^{(i)} - k\hat{e}_d \right) \left(\frac{\partial \hat{h}}{\partial u} \right)^{-1} \quad (15)$$

From Eq. (15), we notice that the control/command available is based on the estimate of tracking error. For example, in case of relative degree one (r=1), the control/command available in Eq. (15) corresponds to a proportional-derivative (PD) feedback. Henceforth, without loss of generality, we assume the sign of control/command effectiveness is positive ($\frac{\partial \hat{h}}{\partial u} > 0$). Using the available control/command from Eq. (14) as control/command correction along with Eq. (13) into Eq. (2) results in

$$\hat{y}_p^{(r)} = \sum_{i=0}^{r-1} a_i \hat{y}_p^{(i)} + b(u_p + \Delta u) + \nu_{ad}(\bar{\mu}) - \nu_{dc}$$

$$= \sum_{i=0}^{r-1} a_i \hat{y}_p^{(i)} + b u_p + \nu_{ad}(\bar{\mu}) - \nu_{dc}$$

$$+ \left[y_d^{(r)} - \left(\sum_{i=0}^{r-1} a_i y_d^{(i)} + b u_p + \nu_{ad}(\bar{\mu}) - \nu_{da} \right) - k(\hat{y}_p - y_d) \right]$$

$$= \sum_{i=0}^{r-1} a_i \hat{y}_p^{(i)} + y_d^{(r)} - \sum_{i=0}^{r-1} a_i y_d^{(i)} - k(\hat{y}_p - y_d)($$

16)

With a definition of the estimate of the safe response tracking error, the above equation can be rewritten into state-space form as follows:

$$\dot{\hat{E}}_d = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 - k & a_1 & \cdots & \cdots & \cdots & a_{r-1} \end{bmatrix} \hat{E}_d , \qquad (17)$$

where $\hat{E}_d = \left[\hat{e}_d \ \hat{e}_d^{(1)} \cdots \hat{e}_d^{(r)} \right]^T$. Equation (17) clearly shows that applying the additional control/command input makes the estimated tracking error dynamics to be asymptotically stable. Therefore, $\hat{y}_p \to y_d$. Now using the error vector definitions, it can be shown that

$$E_d = \hat{E}_d - E , \qquad (18)$$

where $e = \hat{y}_p - y_p$, $E = \left[e \ e^{(1)} \cdots e^{(r)} \right]^T$, $e_d = y_p - y_d$ and $E_d = \left[e_d \ e_d^{(1)} \cdots e_d^{(r)} \right]^T$. The adaptive neural network is designed to guarantee that the error in the estimate of limit parameter dynamics remains ultimately bounded. Furthermore, this bound can be made sufficiently small so as to guarantee that the adaptive estimate of limit parameter dynamics is a reasonable approximation of the true dynamics of the limit parameter. Therefore, assuming that a well-performing adaptive estimate of limit parameter dynamics is available and with an additional control/command input of Eq. (6), we can conclude that the limit parameter tracking error, E_d, will remain bounded.

As a result, the proposed approach for obstacle avoidance and envelope protection can be summarized as given in Fig 8.

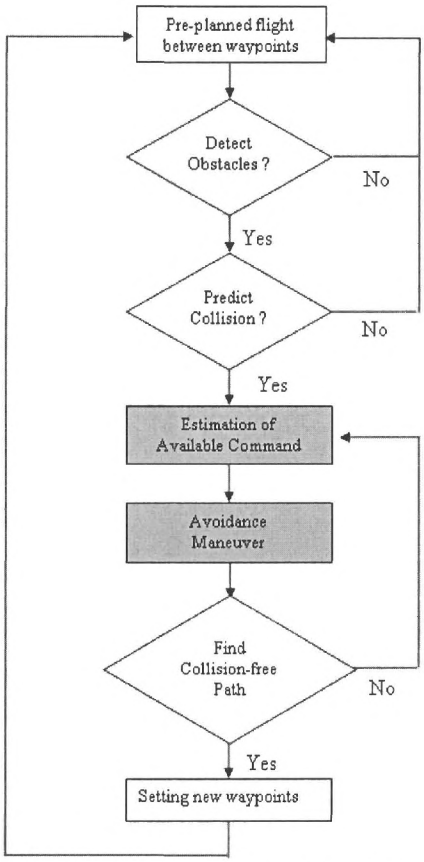


Fig. 8 Decision flow for integrated avoidance

SIMULATION RESULTS

The application of the proposed integrated approach is first demonstrated using a two-dimensional example (Ref. 15). In this example, we consider a load factor limit as an internal constraint. The objective of this limit protection system is to use the full capability of a vehicle as well as to prevent load factor response from exceeding an upper limit. A simplified model is adopted for the UAV trajectory dynamics and a first-order lag transfer function is used for the pitch rate dynamics.

The first step in the design process is to choose a linear approximation model of load factor dynamics. In this example, load factor dynamics is selected from reference 11, which is given by Eq. (19),

$$\dot{\hat{N}}_z = -\left(\hat{N}_z - 1\right) + 4q_c \tag{19}$$

Actual load factor of the vehicle is also computed from acceleration measurement using the following equation

$$N_z = \frac{\|a\|}{g} \tag{20}$$

where $\|\cdot\|$ represents Euclidean norm. The final form of the adaptive estimation used to represent load factor dynamics is given by Eq. (21)

$$\dot{\hat{N}}_z = -\left(\hat{N}_z - 1\right) + 4q_c + \nu_{ad}(\bar{\mu}) - 10\left(\hat{N}_z - N_z\right) \qquad (21)$$

Notice that a static error feedback gain of 10 is used for the linear compensator in Eq. (21). A single hidden layer neural network is used for the adaptive estimation. The normalized input vector to the neural net is used as follows:

$$\bar{\mu} = \left[q_c \quad \frac{N_z}{2} \quad \frac{V_x}{50} \quad \frac{V_y}{50} \right] \qquad (22)$$

The design parameters used in the neural net are provided in Table 1 and Table 2 details a simulation case.

Table 1. Neural network design parameters

Number of neurons	8
Basis function	$\frac{1}{1+e^{-a(x-c)}}$
Output layer learning rate	1.0
Hidden layer learning rate	1.0
E-mod gain	0.05
Sigmoid parameter	$a = 1,\ c = 0$

Table 2. Simulation condition

Initial position of UAV	(0,50)
Speed	50 ft/sec
Load factor upper limit	1.5g
Maximum detection range Rmax	200 ft
Field of view 2φ	60°

Fig. 9 shows the resulting trajectory in the vertical plane using the integrated approach. Even though the UAV detects an obstacle and predicts collision at (200, 50), it has no information on the obstacle configuration. At that point, it starts avoidance maneuver. Available pitch rate command is shown in Fig. 10, which is based on the adaptive estimation of load factor dynamics. When the UAV predicts collision to obstacles, the proposed method applies the maximum available command to be more effective for obstacle avoidance. It seems that the height of the obstacle is assumed to be infinite until the UAV finds out the collision-free path. Fig. 11 shows conditions of obstacle detection and collision prediction during the maneuver. In each figure, '1' means 'Yes', while '0' represents 'No.' Even though obstacle detection lasts for about 6 seconds, collision is predicted for only about 3 seconds. As shown in Fig. 10, the UAV tries to maintain its pitch angle when no more collision is predicted.

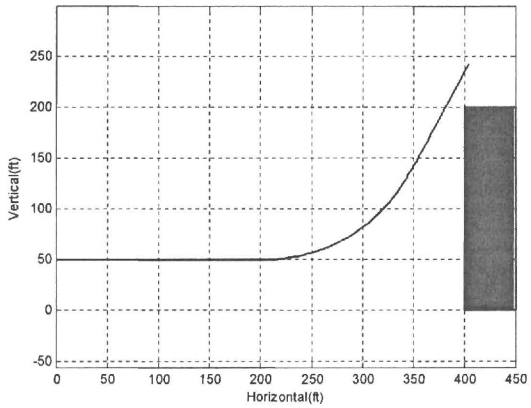


Fig. 9 Avoidance trajectory in a vertical plane

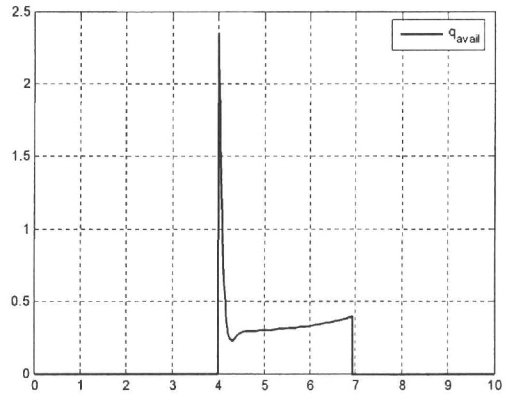


Fig. 10 Available pitch rate command using reactive envelope protection method

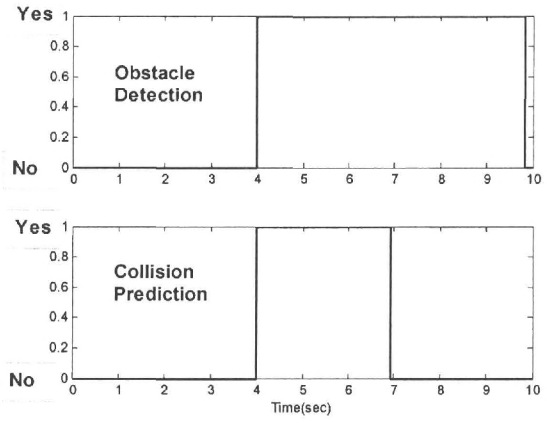


Fig. 11 Obstacle detection and collision prediction

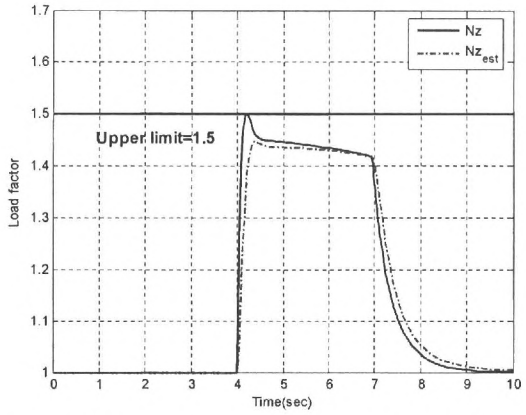


Fig. 12 Adaptive estimation of load factor

The resulting load factor response and its estimate from the adaptive neural net based estimation are shown in Fig. 12. Fig. 13 compares the desired load factor (for limit avoidance) and the actual load factor response. Note that the load factor response stays within the assumed limit of 1.5.

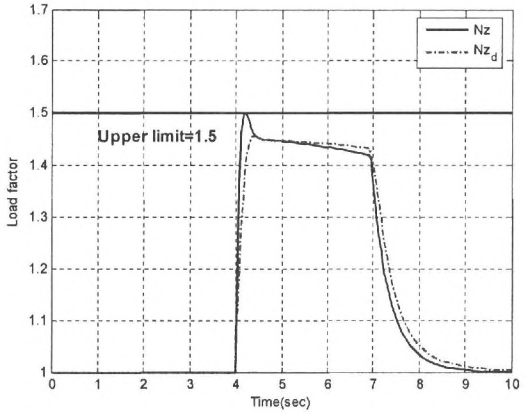


Fig. 13 Desired and actual load factor

Next, we applied the proposed algorithm to a helicopter UAV simulation tool. Fig. 14 shows a picture of Georgia Tech helicopter UAV (GTMax).

Fig. 14 Georgia Tech helicopter UAV GTMax

Contrary to the previous example, a nonlinear flight simulation with 6 degrees of freedom simulation is considered here (Ref. 16). GTMax simulation architecture is shown in Fig. 15. The first component in Fig. 15 is the trajectory generator which provides the position, velocity, and attitude commands to the flight computer as a function of time. The original plan consists of a set of waypoints along with values for nominal velocity. If the obstacle detection radar finds obstacles along the original path, the flight computer starts to generate a new trajectory to avoid obstacles. The default flight controller is an adaptive neural network trajectory following controller. This adaptive neural net controller can be configured as a conventional inverting controller. The flight controller determines actuator commands based on the guidance system commands and the navigation system output. GTMax helicopter model, the helicopter interface model, and sensor models have been developed as a simulation tool. This simulation tool is written primarily in C/C++ and has been developed to allow the test architecture to run on a high-end personal computer.

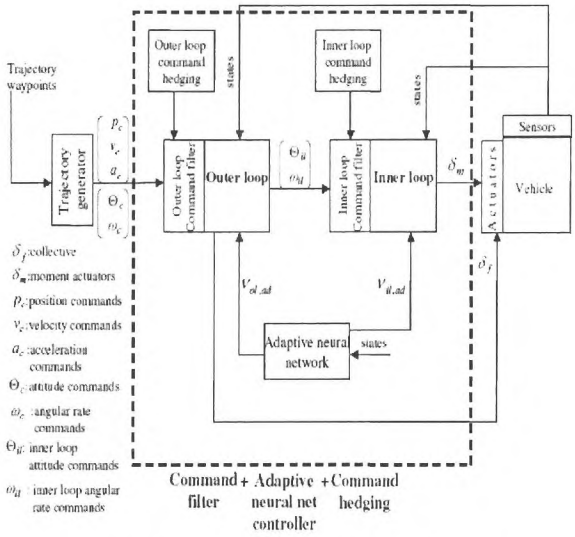


Fig. 15 GTMax simulation architecture

Tables 3 and 4 show the details of neural network design and the simulation condition for GTMax load factor case, respectively.

Table 3 Neural net design for GTMax load factor limiting

Number of neurons	8
Basis function	$\frac{1}{1+e^{-a(x-c)}}$
Output layer learning rate	4.0
Hidden layer learning rate	0.1
E-mod gain	0.02

Sigmoid parameter	$a = 1,\ c = 0$

Table 4 Simulation condition for GTMax load factor limiting

Initial position of UAV	(750, 0, -30)
Destination waypoint	(3100, 0,-30)
Speed	30 ft/sec
Load factor upper limit	1.5g
Maximum detection range Rmax	300 ft
Field of view 2φ	60°

Fig. 16 shows a trajectory in the vertical plane. When the GTMax detects an obstacle and predict a collision at (1700, 0, -30), it starts an avoidance maneuver according to Eqs. (14) and (15). Trajectory commands for the flight computer are generated as follows:

$$\theta_{com}(t) = \int u_{avail}(s)ds \tag{23}$$

$$\dot{x}_{com} = V\cos(\theta_{com}),\ \dot{z}_{com} = -V\sin(\theta_{com}),\ \dot{y}_{com} = 0 \tag{24}$$

$$x_{com} = \int V\cos(\theta_{com}),\ z_{com} = -\int V\sin(\theta_{com}),\ y_{com} = 0 \tag{25}$$

$$\omega_{com} = [0,\ 0,\ 0]^T,\ q_{com} = [1,\ 0,\ 0,\ 0]^T \tag{26}$$

Note that (x, y, z) represents a position in the NED coordinate system and q means a quaternion vector for the attitude representation. Since we consider only a motion in a vertical plane, velocity and position command along the Y axis are set to zero. In addition, Fig. 17 shows the detection condition during the maneuver. '0' represents an obstacle free condition, '1' means an obstacle detected, but no collision predicted condition, and '2' shows a collision predicted condition. If the detection condition says '0', then the UAV tries to fly toward the destination waypoint. Flight path angle obtained by Eq. (23) is shown in Fig. 18. The flight path is commanded as much as possible without violating the load factor limit if the collision to obstacles is predicted. Fig. 19 represents the load factor response, which is maintained within the upper limit.

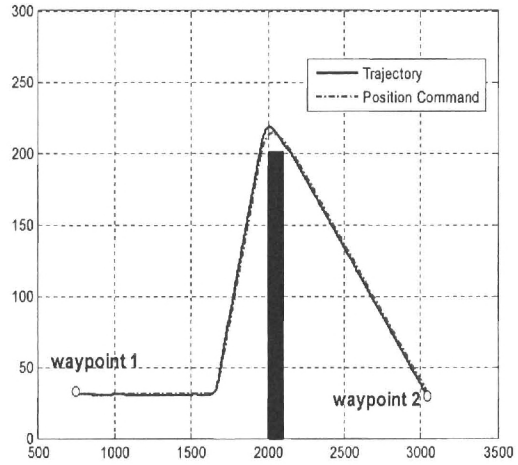


Fig. 16 Trajectory in a vertical plane

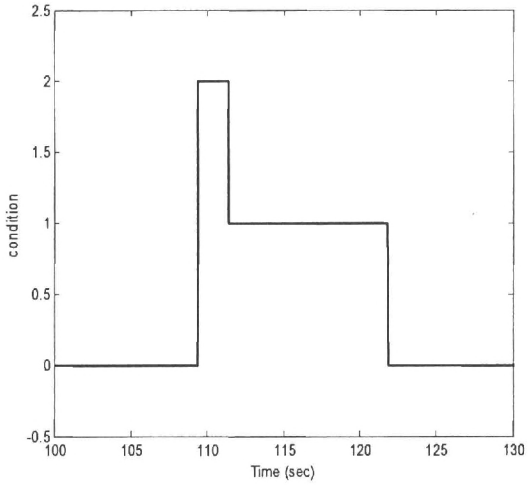


Fig. 17 Flight condition (0=obstacle free, 1=collision free, 2=collision predicted)

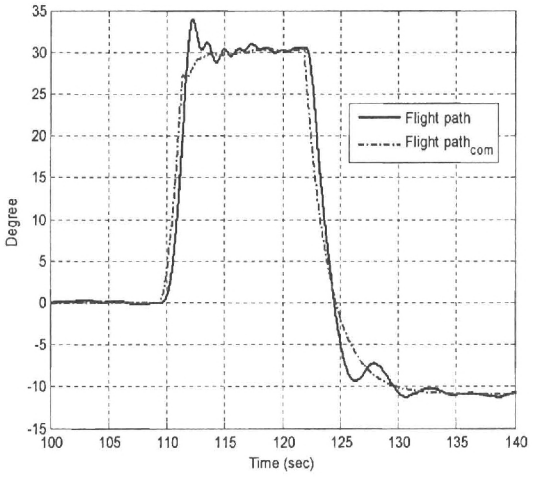


Fig. 18 Flight path command and tracking

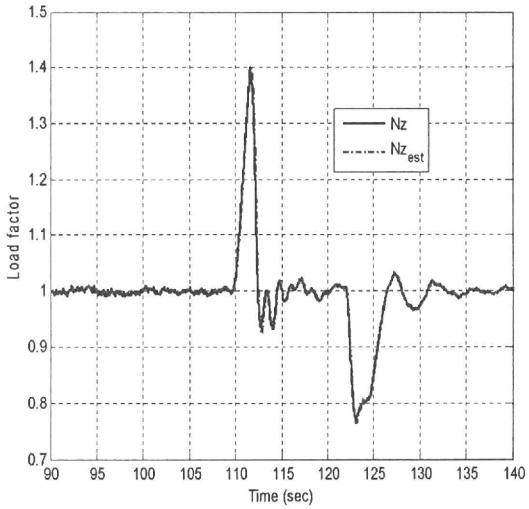


Fig. 19 Load factor response

CONCLUSIONS

Integrated approaches to obstacle avoidance and envelop protection are described in this report. An initial approach for combining reactionary obstacle avoidance and reactionary limit avoidance is discussed by considering a simpler 2-D obstacle avoidance problem of a UAV flying at constant speed. In order to achieve a time optimal solution, the proposed method calculates and applies maximum available command whenever a collision with obstacle is predicted. Simulation results obtained using a simplified model of UAV trajectory and pitch dynamics illustrate the integrated approach studied here. Also, six-degree-of-freedom nonlinear simulation results for the GTMax UAV testbed demonstrate the effectiveness of the proposed approach in arriving at collision-free path while ensuring that the assumed load factor limit is not exceeded.

Future work needs to consider further development of the integrated approach studied in this here, especially in the context of 3-D obstacle avoidance with varying speed. Also, the use of the nonlinear trajectory generation (NTG) method for simultaneous obstacle avoidance and limit avoidance needs to be studied and compared with the integrated reactionary approach.

APPENDIX

Proof of Property 1

From Fig. (6), since

$\overline{AH} = \overline{OA}\cos\theta = \overline{O_3A}\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)$, we can obtain the following $\overline{O_3A} = \dfrac{\overline{AH}}{\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)}$. Thus

$$\overline{O_3B} = \overline{O_3A} - \overline{AB}$$

$$= \frac{\overline{AH}}{\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)} - 2\frac{V}{\dot{\theta}}\sin\left(\frac{1}{2}\dot{\theta}\,\Delta t\right)$$

$$= \frac{\overline{OA}\cos\theta}{\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)} - 2\frac{V}{\dot{\theta}}\sin\left(\frac{1}{2}\dot{\theta}\,\Delta t\right)$$

Similarly, since

$\overline{BH_1} = \overline{O_3B}\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right) = \overline{O_2B}\cos\left(\theta + \dot{\theta}\,\Delta t\right)$, we can obtain $\overline{O_2B} = \overline{O_3B}\dfrac{\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)}$

Thus,

$$\overline{O_2B} = \left\{\frac{\overline{OA}\cos\theta}{\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)} - 2\frac{V}{\dot{\theta}}\sin\left(\frac{1}{2}\dot{\theta}\,\Delta t\right)\right\}\frac{\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)}$$

$$= \frac{\overline{OA}\cos\theta}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)} - 2\frac{V}{\dot{\theta}}\frac{\sin\left(\frac{1}{2}\dot{\theta}\,\Delta t\right)\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right)}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)}$$

Since $2\sin\left(\frac{1}{2}\dot{\theta}\,\Delta t\right)\cos\left(\theta + \frac{1}{2}\dot{\theta}\,\Delta t\right) = \sin\left(\theta + \dot{\theta}\,\Delta t\right) - \sin\theta$, we obtain

$$\overline{O_2B} = \frac{\overline{OA}\cos\theta}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)} - \frac{V}{\dot{\theta}}\frac{\left\{\sin\left(\theta + \dot{\theta}\,\Delta t\right) - \sin\theta\right\}}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)}$$

This equation shows how much distance to obstacle changes during a small time step Δt. Let $\overline{OA}$ and $\overline{O_2B}$ be $r_0(t)$ and $r_0(t + \Delta t)$, respectively. Then $\Delta r_0(t) = r_0(t + \Delta t) - r_0(t)$

$$= r_0\left\{\frac{\cos\theta}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)} - 1\right\} - \frac{V}{\dot{\theta}}\frac{\left\{\sin\left(\theta + \dot{\theta}\,\Delta t\right) - \sin\theta\right\}}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)}$$

An average of distance change over Δt can be expressed as

$$\frac{\Delta r_0}{\Delta t} = \frac{r_0}{\Delta t}\left\{\frac{\cos\theta}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)} - 1\right\} - \frac{V}{\dot{\theta}\,\Delta t}\frac{\left\{\sin\left(\theta + \dot{\theta}\,\Delta t\right) - \sin\theta\right\}}{\cos\left(\theta + \dot{\theta}\,\Delta t\right)}$$

As $\Delta t \to 0$, $\frac{\Delta r_0}{\Delta t}$ becomes $\dot{r}_0(t)$. The first term of the right-hand side becomes $r_0\dot{\theta}\tan\theta$, since

$$\lim_{\Delta t \to 0} \frac{\cos\theta - \cos\left(\theta + \dot{\theta}\,\Delta t\right)}{\Delta t \cos\left(\theta + \dot{\theta}\,\Delta t\right)} = \dot{\theta}\tan\theta$$

Also, the second term of the right-hand side becomes $-V$, since $\lim_{\Delta t \to 0} \frac{\sin\left(\theta + \dot{\theta}\,\Delta t\right) - \sin\theta}{\left(\dot{\theta}\,\Delta t\right)\cos\left(\theta + \dot{\theta}\,\Delta t\right)} = 1$.

Therfore, we obtain a relationship between pitch rate and closing rate as follows:

$$\dot{r}_0 = -V + r_0\dot{\theta}\tan\theta$$

REFERENCES

[1] Bone, E., and Bolkcom, C., *Unmanned Aerial Vehicles: Background and Issues*, Novinka Books, 2004.

[2] Von Stryk, O, and Bulirsch, R, "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, 1992, pp.357-373.

[3] Chen, Y., and Huang, J., "A New Computational Approach to Solving a Class of Optimal Control Problems," *International Journal of Control*, Vol.58, 1993, pp.1361-1383.

[4] Charlet, B., Levine, J., and Mario, R., "On Dynamic Feedback Linearization," *Systems and Control Letters*, Vol.13, 1989, pp. 143-151.

[5] Fliess, M,. Levine, J, Martin, P., and Rouchon, P., "Flatness and Defect of Nonlinear Systmes: Introductory Theory and Examples," *International Journal of Control*, Vol. 61, 1995, pp. 1327-1360.

[6] Jeram, G., "Open Design for Helicopter Active Control Systems," *AHS 58th Annual Forum*, Montreal, Canada, June, 2002, pp. 2435-2454.

[7] Yavrucuk, I., *Adaptive Limit Margin Detection and Limit Avoidance*, PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, 2003.

[8] Horn, J., *Flight Envelope Limit Detection and Avoidance*, PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, 1999.

[9] Cheng, H. L., and Lam, T., "Automatic Guidance and Control for Helicopter Obstacle Avoidance," Journal of Guidance, Control, and Dynamics, Vol. 17, No. 6, pp. 1252-1259.

[10] Howlett, J., and et al, "Flight Evaluation of a System for Unmanned Rotorcraft Reactive Navigation in Uncertain Urban Environments," *AHS 63rd Annual Forum*, Virginia Beach, Virginia, May, 2007.

[11] Unnikrishnan, S., Yavrucuk, I. and Prasad, J.V.R., "Reactionary Envelope Protection for Autonomous UAVs," Proceedings of the 61st American Helicopter Society Annual Forum, Grapevine, TX, June 1-3, 2005.

[12] Unnikrishnan, S. and Prasad, J.V.R., "Flight Evaluation of a Reactionary Envelope Protection System for UAVs," *AHS 62nd Annual Forum*, Phoenix, Arizona, May, 2006.

[13] Unnikrishnan, S., Jeram, G. and Prasad, J.V.R., "Optimal Control Formulation and Nonlinear Trajectory Generation for Envelope Protection," Proceedings of the 61st American Helicopter Society Annual Forum, Grapevine, TX, June 1-3, 2005.

[14] Reynold, C., "Not Bumping into Things," Notes on obstacle avoidance for the course on Physically based modeling at SIGGRAPH 88 available online @ http://www.red3d.com/cwr/nobump/nobump.html, data accessed September 2007.

[15] Prasad, J.V.R., Moon, J. and Kim, C-J., "Obstacle Avoidance with Envelope Protection for UAVs," Paper presented at the International Forum on Rotorcraft Multidisciplinary Technology, Seoul, Korea, October 15-16, 2007.

[16] Moon, J., Prasad, J.V.R. and Kim, C-J., "Reactive Obstacle Avoidance for Autonomous UAVs: An Envelope Protection Based Approach," Accepted for presentation at the 2008 AHS Forum, Montreal Canada, May 2008.

Appendix III

Collaborative Project
Between Georgia Institute of Technology and
Konkuk University
For
Development of an
Integrated Product and Process Development (IPPD) Framework
for Rotorcraft Preliminary Design and Multidisciplinary Design Optimization (MDO)
First Year Results

Dr. Daniel P. Schrage
Professor and Director, CERT and CASE
School of Aerospace Engineering, Georgia Tech

I. Rationale for Collaborative Project

The necessary rotorcraft preliminary design and development process, based on an Integrated Product and Process Development (IPPD) approach, is illustrated below in Figure 1. It is separated into Product and Process Development and illustrates two design iteration loops (Conceptual and Preliminary), an Initial Product Data Management Loop, and a Process Design Iteration Loop. Product Life-cycle Management (PLM) tools (CATIA, ENOVIA and DELMIA) are also indicated, along with disciplinary analysis tools as well. The three year collaborative project will support KKU in creating and executing this environment for conducting Multidisciplinary Design Optimization (MDO) for current and future rotorcraft designs. It was determined that collaboration for developing this IPPD Framework will be enhanced through a joint GT-KKU graduate team working together on a proposal in response to the 2007 AHS International Student Design Competition for a SMART-ROTOR helicopter that minimizes energy consumption..

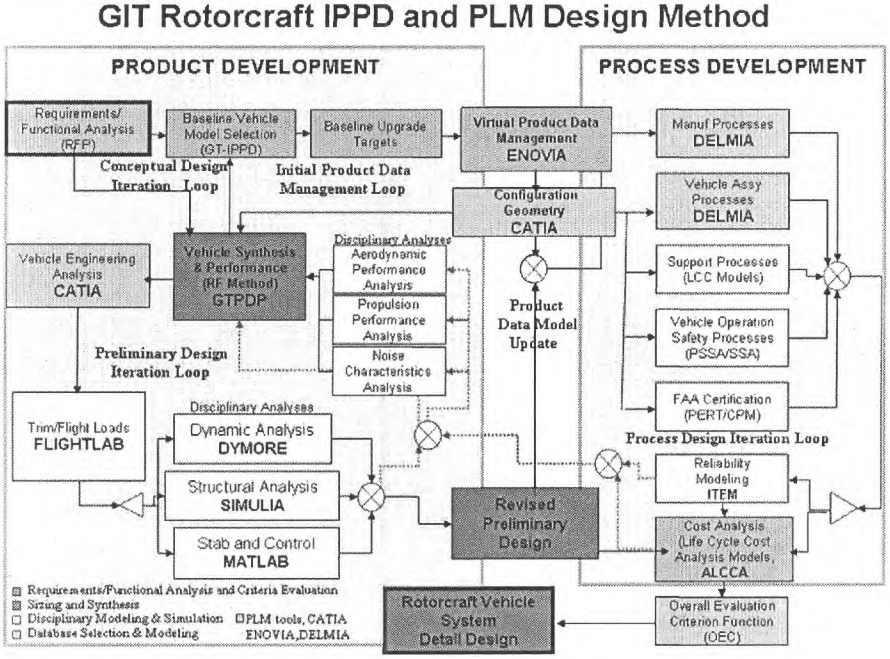


Figure 1. IPPD Framework for Rotorcraft PD and MDO

II. First Year Results

The first year was spent identifying the necessary approach for developing the framework and identifying the necessary multidisciplinary (MD) tools and integrating MD optimization (MDO) architecture for the Preliminary Design Loop Iteration illustrated in Figure 1. The following results have been achieved:

1. It was determined that the best approach for developing the initial framework would be to form a multidisciplinary (MD) team of Georgia Tech and Konkuk University that could help develop the initial framework as part of their entry as a MD team for the AHS/Eurocopter sponsored 25th Student Design Competition Request for Proposal (RFP) for a SMART-ROTOR for Minimizing Energy Consumption. A MD team of Georgia Tech and Konkuk University students has been formed and all of the team members are currently taking the Georgia Tech Rotorcraft Design I Course taught by Dr. Schrage during the 2007 Fall Semester. This team is currently completing the Conceptual Design Iteration Loop in Figure 1 and will meet at Georgia Tech for two weeks in January 2008 to commence initiation of the Preliminary Design Iteration Loop (PDIL)

2. A survey of MD tools that can be utilized for conducting the PDIL has been conducted. Several MD tools have been identified for exercising the PDIL the SMART-ROTOR. They are:

   i. A customized fuel balance method (R_F) developed by the MD team or as part of other student's Masters and PhD theses will be utilized in conjunction with the Georgia Tech Preliminary Design Program (GTPDP), which will also be used for verification and validation and further development of the R_F Method.

   ii. The choice of the Computational Fluid Dynamics (CFD) tool to be utilized has been based on the advanced CFD tools developed by Dr. Lakshmi Sankar. It has been determined that the Rotor Design Computer Code, GTROT3D, 3-D Navier-Stokes solver for modeling rotors in hover and in forward flight will be used. A major decision on the choice of this tool was the requirement in the AHS/Eurocopter RFP to assess a morphed rotor, thus requiring the use of a high fidelity CFD tool.

   iii. The choice of the Computational Structural Dynamics (CSD) tool is to use the DYMORE multi-body dynamic analysis tool developed by Dr. Olivier Bauchau. This is based on the extensive experience that Dr. Sankar and Dr. Bauchau have had in integrating CFD-CSD tools and on the need to assess a morphed rotor.

   iv. For Flight Dynamics and Controls modeling it has been determined to use the FlightLab computer code, a Commercial Off The Shelf (COTS) software tool developed by ART, Inc, which is currently available at Georgia Tech and Konkuk University. To enhance carefree maneuvering and flight into urban canyons, as required for the SMART-ROTOR, the Integrated Envelope Protection Program (IntEPP) developed by Dr. JVR Prasad has been selected. This program provides computer a code for integrated obstacle avoidance with envelope protection, based on a reactive trajectory generation method.

3. For the MDO Integrating Architecture two different approaches are being evaluated: the **Model Center** approach offered by Phoenix Integration and the P.I.A.n,O architecture, jointly developed at Hanyang University and Georgia Tech and offered by FRAMAX. Model Center has been used in previous rotorcraft MDO research by Adeel Khalid[1] because of its flexibility to incorporate several existing commercial packages e.g. Excel, Matlab, MathCad, CATIA etc. ModelCenter also facilitates the use of wrappers to integrate in house legacy codes. Khalid's MDO integration approach was to provide an All At Once (AAO) MDO Optimization Loop for minimizing weight and cost in the PDIL illustrated in Figure 2. The Model City environment is illustrated in Figure 3.

4.

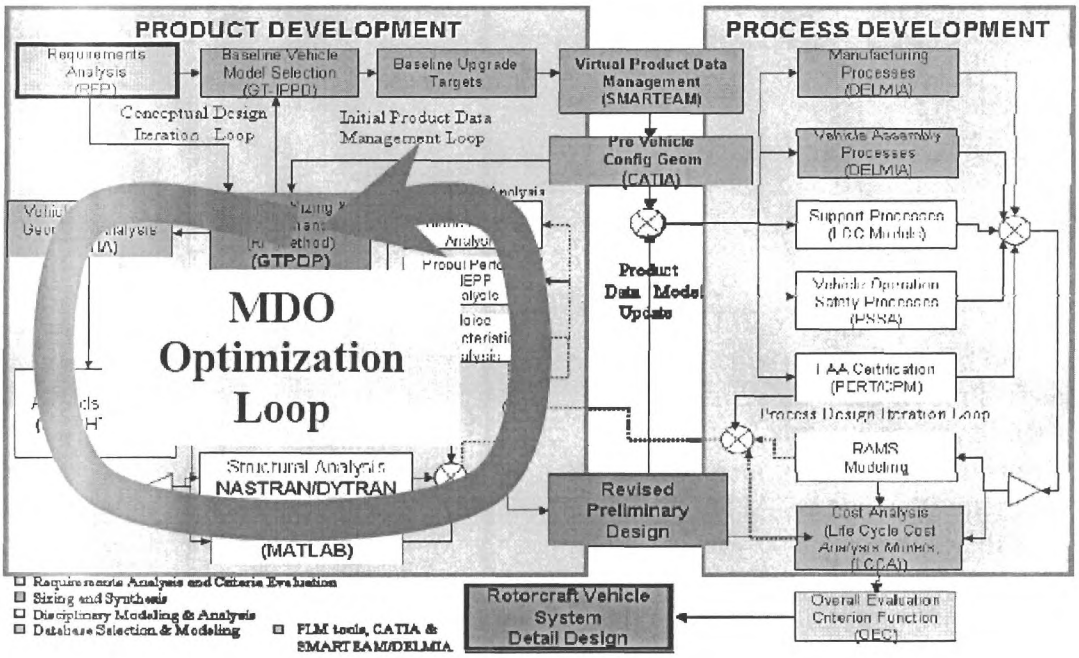


Figure 2. MDO Integration Approach for AAO MDO Optimizaition

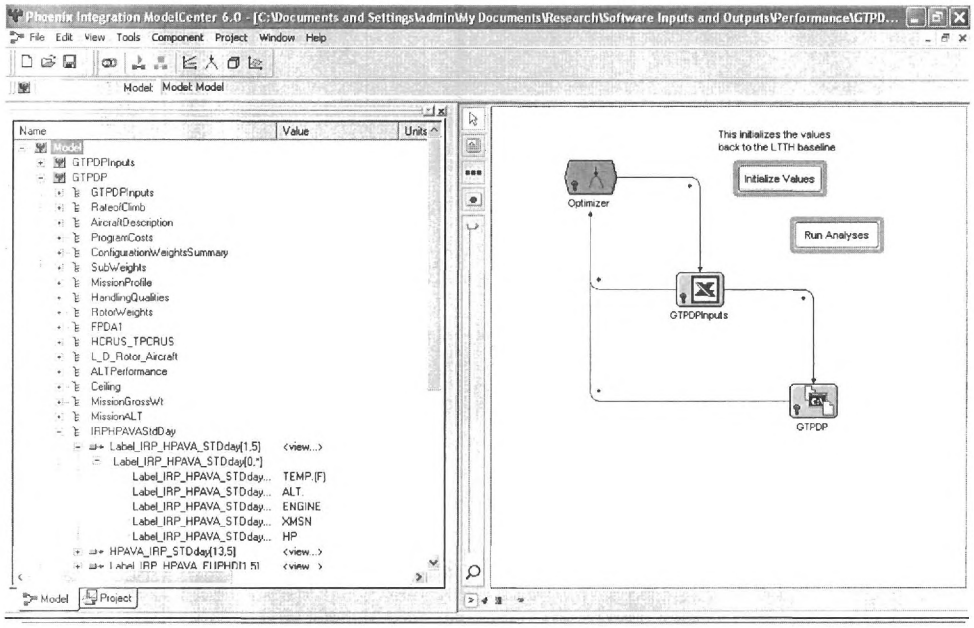


Figure 3. The Model Center MDO Integration Architecture

The second MDO integrating architecture that is being evaluated is **P.I.A.n.O.**, the abbreviation for Process Integration, Automation and Optimization. It is a PIDO(Process Integration and Design Optimization) tool for executing numerous design methodologies, taking into consideration coupled analysis in various fields. An illustration of the PIDO integration approach is illustrated in Figure 4. The application of P.I.A.n.O for coupled rotorcraft systems MDO is illustrated in Figure 5 and was presented at ROTOR-KOREA. Continued evaluation of both MDO integrating architecture approaches is taking place the rest of this calendar year.

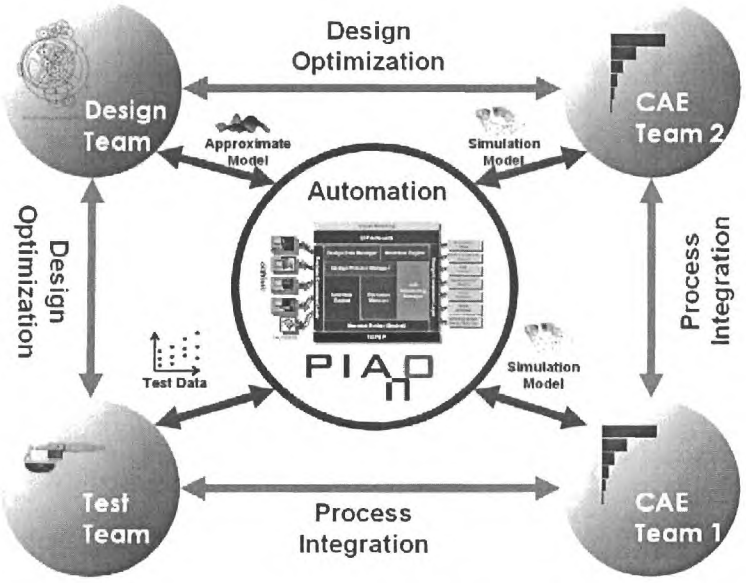


Figure 4. PIAnO PIDO Framework

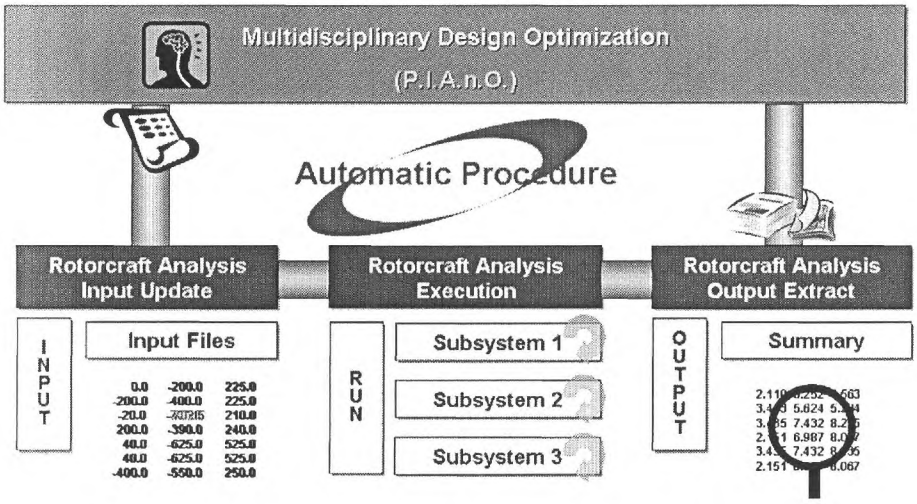


Figure 5. PIAnO for Coupled Rotorcraft Systems MDO

In addition to using MDO for the Product Development side of the IPPD framework illustrated in Figure 1, the use of an integrated Product Life-cycle Management (PLM) environment is also be pursued. Both Georgia Tech and Konkuk University have access to the CATIA V5 computer aided design (CAD) software package that can be used for Virtual Product development. In addition, Georgia Tech is signing a collaborative agreement with Dassault Systemes to have access to most of the rest of their integrated PLM toolset. As can be seen in Figure 6, Dassault Systemes has developed life cycle set of integrated PLM tools and that ENOVIA Virtual PLM (VPLM) provides the primary means for PLM collaboration. It is also illustrated in the Figure 1 that ENOVIA database management coupled with CATIA V5 provides the linkage between

Product and Process Development. With this in mind, it is hoped to use a PLM environment to further the collaboration between Georgia Tech and Konkuk University.

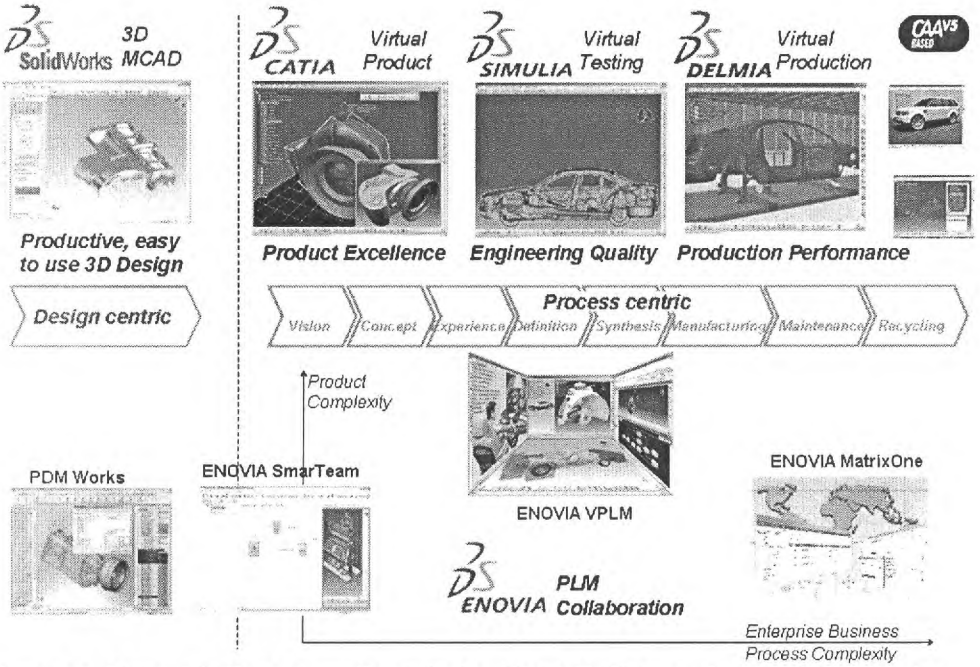


Figure 6. Dassault Systemes Brands for PLM Collaboration and Integration

III. Plans for Years Two and Three

The plan for year two is to complete the initial IPPD Rotorcraft Preliminary Design framework utilizing the joint Georgia Tech – Kon Kuk University graduate student design team responding to the AHS/Eurocopter SMART-ROTOR, as well as additional researchers from both Georgia Tech and Kon Kuk University. The following are specific actions planned:

1. Have the KKU students spend two weeks at Georgia Tech in January 2008 to begin initiating the PDIL in Figure 1. This will constitute In Process Review (IPR) #1.
2. Have a Georgia Tech researcher, post doc or research engineer, spend six months or longer at KKU as a visiting scholar to assist in the MDO development, starting in January 2008
3. Have In Process Review (IPR) #2 at KKU for approximately one week in March 2008 with members of the GT research team being present.
4. Have In Process Review (IPR) #3 at KKU for approximately one week in May 2008 to assess the status of completion of the initial IPPD Rotorcraft Design framework and prepare the necessary material for review. The GT research team will be present.

Once the initial IPPD Rotorcraft Preliminary Design framework is completed in response to the AHS/Eurocopter RFP for a SMART-ROTOR for Minimizing Engineering

Consumption, the next IPPD application will be for a rotorcraft of interest to Kon Kuk University. This effort will be accomplished during Year Three.